

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-295840  
(43)Date of publication of application : 10.11.1995

(51)Int.Cl. G06F 9/46

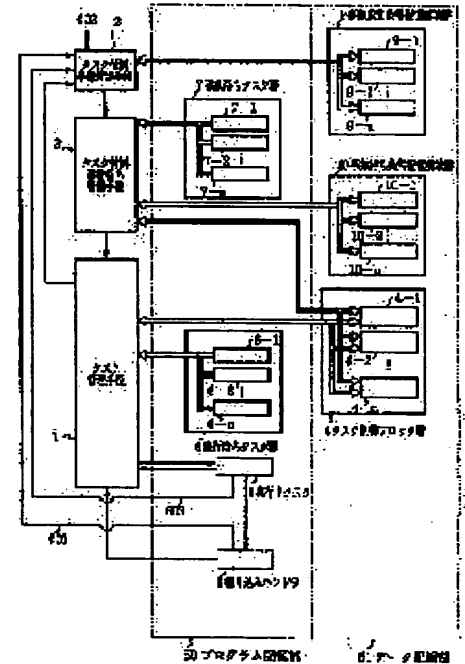
(21)Application number : 06-090879 (71)Applicant : NEC IC MICROCOMPUT SYST LTD  
(22)Date of filing : 28.04.1994 (72)Inventor : HANESAKA YOSHINORI

## (54) EVENT MANAGEMENT SYSTEM FOR MULTIPROGRAMMING SYSTEM

### (57)Abstract:

**PURPOSE:** To improve the using efficiency of resources and also to improve the productivity and the maintenance performance in a multiprogramming system by decreasing the storage areas and the program steps used by the OS of the system which is applied to a built-in system.

**CONSTITUTION:** An event management system includes a task management means 1, a task attached event management means 2, a task attached event wait management means 3, a task control block group 4, an under-execution task 5, an execution wait task group 6, an event wait task group 7, an interruption handler 8, an event establishment condition storage area group 9, and an event wait condition storage area group 10. The information included in the task 5, the groups 6 and 7, and the handler 8 are stored in a program storage part 50. Then the information included in the groups 4, 9 and 10 are stored in a data storage part 51.



## LEGAL STATUS

[Date of request for examination]	28.04.1994
[Date of sending the examiner's decision of rejection]	18.02.1997
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3245500
[Date of registration]	26.10.2001
[Number of appeal against examiner's decision of rejection]	09-04282
[Date of requesting appeal against examiner's decision of rejection]	19.03.1997
[Date of extinction of right]	



(19) 日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11) 特許番号

特許第3245500号  
(P3245500)

(45) 発行日 平成14年 1 月15日 (2002. 1. 15)

(24) 登録日 平成13年10月26日 (2001. 10. 26)

(51) Int.Cl.<sup>7</sup>

G 0 6 F 9/46

識別記号

3 4 0

F I

G 0 6 F 9/46

3 4 0 B

請求項の数 1 (全 18 頁)

(21) 出願番号 特願平6-90879  
(22) 出願日 平成 6 年 4 月28日 (1994. 4. 28)  
(65) 公開番号 特開平7-295840  
(43) 公開日 平成 7 年11月10日 (1995. 11. 10)  
審査請求日 平成 6 年 4 月28日 (1994. 4. 28)  
審判番号 平9-4282  
審判請求日 平成 9 年 3 月19日 (1997. 3. 19)

(73) 特許権者 000232036  
エヌイーシーマイクロシステム株式会社  
神奈川県川崎市中原区小杉町 1 丁目403  
番53  
(72) 発明者 羽坂 佳典  
神奈川県川崎市中原区小杉町一丁目403  
番53 日本電気アイシーマイコンシステ  
ム株式会社内  
(74) 代理人 100082935  
弁理士 京本 直樹 (外 2 名)

合議体

審判長 西川 正俊

審判官 長島 孝志

審判官 大橋 隆夫

最終頁に続く

(54) 【発明の名称】 マルチプログラミングにおける事象管理方式

(57) 【特許請求の範囲】

【請求項 1】 プログラムを細分化したタスクごとに設定される当該タスクの実行可能な条件（以下、事象と云う）の種別と、当該種別に対応する待ち内容と、条件が成立した結果からなる成立条件を格納する事象成立条件記憶手段と、  
使用頻度の高い特定事象の種別と当該種別に対応する待ちの内容とを格納する事象成立条件テーブルと、  
前記事象成立条件記憶手段に格納されている事象が成立した時点に対応する事象待ち解除条件を格納する事象待ち条件記憶手段と、  
前記タスクを制御するための制御情報を格納するタスク制御ブロックと、  
実行中のタスクにおいて事象待ちが発生した場合、該事象が事象成立条件テーブルにあるときは事象成立条件テ

ーブルから登録内容を読み出して事象成立条件記憶手段に書きこみ、事象成立条件テーブルにないときは、発生した事象の成立条件を事象成立条件記憶手段に書きこむものであり、また、タスクの事象が発生するたびに、当該事象を前記事象成立条件記憶手段に格納されている事象成立条件と比較照合して、対象とする事象の成立の可否を検出するタスク付属事象管理手段と、  
実行中のタスクにおいて事象待ちが発生した場合、当該タスクを前記タスク制御ブロックを使用して事象待ち状態に遷移させ、また、当該事象待ち状態のタスクに対する前記事象待ち解除条件が成立した時点において、当該タスクを実行待ち状態に遷移させるタスク付属事象待ち管理手段と、  
前記タスク付属事象待ち管理手段により、事象待ち状態に遷移されたタスクを一時的に格納する事象待ちタスク

(2)

3

記憶手段と、

前記タスク付属事象待ち管理手段により、実行待ち状態に遷移されたタスクを一時的に格納する実行待ちタスク記憶手段と、

所定のCPU（中央処理装置）において実行されているタスクを格納している実行中タスク記憶手段と、

前記実行中のタスクと、前記実行待ちタスク記憶手段に格納されている実行待ちタスクとの内より一つのタスクを選択して、前記タスク制御ブロックを使用して実行状態に遷移させるタスク管理手段と、

を少なくとも備えることを特徴とするマルチプログラミングにおける事象管理方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はマルチプログラミングにおける事象管理方式に関し、特にマイクロコンピュータと周辺ハードウェアとのインタフェースを必要とする電話およびファクシミリ等において、複数のタスクを事象ごとに切替えるOSと、当該OS上において動作するアプリケーション・プログラムとを、ROM等のファームウェアとして記憶して組み込むシステムに適用されるマルチプログラミングにおける事象管理方式に関する。

【0002】

【従来の技術】従来、複数のタスクを並行動作させる管理プログラムとして機能するマルチプログラミングにおける事象管理方式（以下、マルチプログラミング・システムと云う）は、複雑な制御用システムを小さな仕事の単位であるタスクとして細分化するとともに、当該タスク間の関係およびタスクと入出力（外部システム）との関係を明確にすることにより、生産性と保守性の向上を図るために用いられている。例えば、CQ出版社の別冊インタフェース・ブートストラップNO. 6、同じくCQ出版社のインタフェース1992年12月号、日本電気（株）の78K/0シリーズ用リアルタイムOS RX78K/0基礎編 資料番号EED-912A（第2版）および日本電気（株）の78K/3リアルタイムOSアプリケーション 資料番号EEA-611A（第2版）等々に示されているように、一つしかないCPUに対して複数のタスクを割り当てて、時間および外部システムからの入力等をタスク切替えの要因（事象）として使用し、タスク一つ一つに対するCPU割り当て時間を分配することにより、前記管理プログラムを実現している。このような管理プログラムは、マルチタスクOSまたはリアルタイムOSとも呼ばれている。

【0003】図14は、従来のOSを用いた一例を示すマルチプログラミング・システムの内部構成図である。図14に示されるように、本従来例は、タスク管理手段1と、タスク制御ブロック4-1、4-2、……、4-nを含むタスク制御ブロック群4と、実行中タスク5と、実行待ちタスク6-1、6-2、……、6-nを含

4

む実行待ちタスク群6と、実行待ちタスク7-1、7-2、……、7-nを含む実行待ちタスク群7と、割り込みハンドラ8と、メール・ボックス12-1、イベント・フラグ12-2、時間待ちタイマ12-3および起床要求カウンタ12-4を含む事象状態記憶領域12と、事象待ち管理手段13とを備えて構成される。なお、実行中タスク5、実行待ちタスク群6、事象待ちタスク群7および割り込みハンドラ8は、当該システム全体を制御するためのプログラムの集合であり、これらは全てプログラム記憶部50内に格納されている。このプログラム記憶部50の例としてはROM等が用いられている。また、タスク制御ブロック群4および事象状態記憶領域12に、プログラム記憶部50に格納されているタスクを制御するための情報が格納されるデータ記憶部51に包含されている。このデータ記憶部51の例としてはRAM等が用いられる。

【0004】図14において、実行中タスク5は、プログラム記憶部50に格納されているプログラムであり、現在時点において、CPUにおいて処理されているプログラムの内容を意味している。この実行中タスク5を処理している状態は実行状態（RUN状態）と呼ばれている。実行待ちタスク群6は、いつでも実行することができる状態にあるが、前記CPUが、実行中タスク5より解放されるのを待機しているプログラムであり、この待機状態は、実行待ち状態（READY状態）または実行可能状態と呼ばれている。事象待ちタスク群7は、或る事象が成立することを待機しているプログラムであり、この待機状態は、事象待ち状態（WAIT状態）と呼ばれている。また、割り込みハンドラ8は、当該システム全体を制御するためのプログラムの集合体を意味しており、プログラム記憶部50に格納されている。

【0005】タスク制御ブロック群4と事象状態記憶領域12には、プログラム記憶部50を制御するための情報が格納され、データ記憶部51に包含されて構成されており、それぞれRAM等により形成される。タスク制御ブロック4は、タスクの実行が中断された時には、再度、当該タスクの続きを実行することが可能な範囲の情報が格納されており、TCBとも呼ばれている。プログラム記憶部50内の実行待ちタスク群6および事象待ちタスク群7は、タスク制御ブロック4に含まれているタスク持続アドレスにより管理され、一つのグループにまとめられたプログラムであり、その構造は待ち行列（キュー）と呼ばれている。事象状態記憶領域12は、メール・ボックス12-1、イベント・ボックス12-2、時間待ちタイマ12-3および起床要求カウンタ12-4により形成されており、事象の条件であるメッセージ、イベント、時間および起床要求等の現在の状態を示す情報が格納されている領域である。

【0006】タスク管理手段1および事象待ち管理手段13は、プログラム記憶部50に格納されている、実行

(3)

5

中タスク5、実行待ちタスク群6および事象待ちタスク群7等を含むタスクならびに割り込みハンドラ8を、データ記憶部に格納されている情報を基に制御する機能を有する手段であり、この内のタスク管理手段1は、実行状態にある実行中タスク5と、実行待ちタスク群6に含まれているタスクの中から、或る取り決めにより一つのタスクを選び出して、実行状態に移行させる機能を有している。このタスク管理手段1による上記の動作は、スケジューリングと呼ばれており、また当該タスク管理手段1自体はスケジューラと呼ばれる。今、一例として、タスク管理手段1により、実行待ちタスク群6から一つのタスクが選び出された場合には、現在、実行状態にあった実行中タスク5は実行待ち状態に遷移し、待ち状態にあった実行待ちタスク群6の内の一つのタスクが実行状態に遷移する。このタスクの遷移状態（切替状態）は、ディスパッチと呼ばれており、実行中タスク5が中断されていた時のレジスタ等の状態を保存する動作（コンテキスト保存）と、実行待ち状態であったタスクが中断された時のレジスタ等の状態を復元する動作（コンテキスト復元）とを合わせた状態を意味している。タスク管理手段1のスケジューリングの方法としては、タスクごとに優先順位を与えて、当該優先順位の高いものから順に実行する優先順位実行方式と、実行待ち状態になった順に実行する先着実行方式と、タスクごとに実行可能の時間を設けてその時間が経過した時に切替えるタイムスライス方式等があるが、これらの方式を適当に組み合わせる用いることが多い。なお、以後の説明においては、優先順位方式と先着順方式とを組み合わせる行う方式を例として説明するものとする。

【0007】実行中タスク5より出力される事象制御情報603は、実行状態のタスクから他のタスクに対して送信される信号またはデータ等であり、1ビットのフラグ情報を持つイベント・フラグおよび1バイト以上のデータ群であるメッセージ等が含まれており、タスク間の同期をとるために使用される。割り込みハンドラ8より出力される事象制御情報601は、事象を設定する信号およびデータを含む信号であり、タスク管理手段1に入力されて、当該タスク管理手段1による制御作用に関与する。事象待ち管理手段13に入力される事象制御情報602は、外部回路またはCPU（図示されない）のタイマから入力されるタイマ割り込み信号等を含む信号であり、事象待ち管理手段13におけるOSの時間管理用の基準タイマとして使用される。事象待ち管理手段13は、上記の事象制御情報601、602および603により、事象待ちタスク群7の中から事象待ちを解除するタスクを選択して、実行待ちタスク群6に移動させる手段である。この事象待ちを解除するタスクを実行待ちタスク群6に移動させる際には、前記タスク管理手段1において行われたスケジューリングと同等のスケジューリングが行われる。

6

【0008】マルチプログラミング・システムのシステム全体としては、通常、タスク管理手段1により、実行中タスク5と実行待ちタスク群6とを頻繁に切替えることにより動作が行われている。しかしながら、このようにタスクの切替えが頻繁に行われると、コンテキストの保存と復元に多くの時間を要する状態となり、一定時間当りのタスク実行時間が短縮される結果となる。この問題に対処するために、或る事象が発生するまで、タスクを切替えの対象外とする機能を持った事象待ち管理手段13が設けられている。

【0009】以下、図14および図15（a）を参照して、実行中タスク5が、事象待ち管理手段13により事象待ち状態に移行する動作について説明する。まず、実行タスク5において事象待ちが発生した場合には、事象待ち管理手段13による制御作用に移行して、ステップB1においては、登録内容を事象状態記憶領域群12に書き込む。次いでステップB2においては、実行中タスク5を事象待ちタスク群7に移動させて、当該実行中タスク5を事象待ち状態にする。実行中タスク5の移動終了後には、タスク管理手段1に制御作用が移りスケジューリングが行われて、実行待ちタスク群6の中から優先順位の高いタスクが実行中タスク5に移行する。

【0010】次に、図14および図15（b）を参照して、事象待ちタスク群7のタスクが、事象待ち管理手段13により実行待ち状態に移行する動作について説明する。タイマ割り込みにより発生した事象制御情報602、または、実行中タスク5および割り込みバンドラ8においてそれぞれ発生した事象制御情報603および601によって、事象待ち管理手段13に制御が移り、ステップB3においては、発生した事象を待つタスクが事象待ちタスク群7に存在するか否かを調べる。ステップB3の判断処理において、当該タスクが事象待ちタスク群7に存在しない場合には、タスク管理手段1に制御が戻り処理は終了する。また、当該タスクが事象待ちタスク群6に存在する場合には、ステップB4において、事象待ちタスク群7に含まれている対象事象を実行待ちタスク群5に移動させて実行待ち状態にする。当該対象事象の移動後においてはタスク管理手段1に制御が移り、スケジューリングが行われて、実行待ちタスク群6の中から優先順位の高いタスクが実行中タスク5に移動する。

【0011】図16は、それぞれ従来のOSを使用した時の、実行待ち行列45、時間待ち行列46、イベント待ち行列47およびメッセージ待ち行列48を含む事象待ち行列の形成例である。ポインタ400、402、404および406は、それぞれ、これらの実行待ち行列45、時間待ち行列46、イベント待ち行列47およびメッセージ待ち行列48を含む事象待ち行列の先頭のタスク制御ブロックのタスク接続アドレスとして定義されるポインタである。これらのポインタは、待ちタスクが

(4)

7

存在しない場合には、例えばヌルポインタ等のあり得ないアドレスに設定される。またポインタ401<sub>1</sub>、401<sub>2</sub>、401<sub>3</sub>、403<sub>1</sub>、403<sub>2</sub>および405は、次のタスク制御ブロックのポインタである。この内、ポインタ401<sub>2</sub>、403<sub>2</sub>および405はターミネータであり、例えば、ヌルポインタ等のあり得ないアドレスに設定される。図14の従来のマルチプログラミング・システムの内部構成図において、事象待ち管理手段13により、図16の時間待ち行列46、イベント待ち行列47およびメッセージ待ち行列48を含む事象待ち行列のタスクが、図16に示される実行待ち行列45に移動する際には、このポインタを書き変えることにより実現されている、また、実行中のタスクが、時間待ち行列46、イベント待ち行列47およびメッセージ待ち行列48を含む事象待ち行列のタスクに移動する時には、時間待ち行列46、イベント待ち行列47、またはメッセージ待ち行列48の先頭または後尾のポインタを書き変えることにより実現されている。

【0012】次に、図17に示されるOSの制御構造例と合わせて動作について説明する。アプリケーション・プログラムの観点から見ると、タスク管理手段1と事象待ち管理手段13との間には、システム・コール14と呼ばれるOS制御用の関数が、アプリケーションプログラムとタスク管理手段1／事象待ち管理手段13との間を取り持っている。例えば、図14の実行待ちタスク群6に対して、新たにタスクを追加したい場合には、まず、タスク生成用システム・コールをすることによりタスク制御ブロック群4を生成し、その後、タスク起動システム・コールを発行することにより、実行待ちタスク群6の最後にタスクを追加する。また、起動したタスクの優先順位を変更したい時には、優先順位変更用のシステム・コールを発行することにより、タスク制御ブロック群4に含まれている優先順位番号を変更する。このように、現在時点における実行中タスク5の中で、発行されたシステム・コール、またはOS自身が管理している事象制御情報602により、タスク管理手段1と事象待ち管理手段13に制御が移る。タスク管理手段1においては、実行待ちタスク群6の優先順位の高いタスクから順次切替えおよび実行が繰返して行われる。もしも、事象待ち管理手段13により、新たに事象待ちタスク群7から実行待ちタスク群6にタスクが移動してきた場合には、当該タスクが、実行中タスク5のタスクよりも優先順位の高いものであれば、実行中のタスクが中断されて、移動してきたタスクが実行される。

【0013】事象状態記憶領域12には、図16において説明したメッセージ待ち行列48の情報を格納するメール・ボックス12-1と、イベント待ち行列47の情報を格納するイベント・フラグ12-2と、時間待ち行列46の情報を格納する時間待ちタイマ12-3と、起床待ち行列の情報を格納する起床要求カウンタ12-4

8

が、少なくとも存在している。以下においては、従来例における問題点に関連して、上記のメール・ボックス12-1、イベント・フラグ12-2、時間待ちタイマ12-3および起床要求カウンタ12-4を、それぞれ利用する場合の事象待ち管理手段13の動作について説明する。

【0014】メール・ボックス12-1を利用する場合の事象待ち管理手段13の動作について説明する。まず、実行中タスク5の中から事象情報603が事象待ち管理手段13に伝達され、これを受けて事象待ち管理手段13において、当該事象情報603の内容、この場合にはメッセージが伝達されているものと判断されると、事象待ち管理手段13により、メール・ボックス12-1に伝達されたメッセージが、当該メール・ボックス12-1に書き込まれる。更にまた、当該メール・ボックス12-1内にある事象待ち行列、即ち事象待ちタスク群7のメッセージ待ちをしているタスクの情報の中に、事象待ちタスク7-1、7-2、……、タスク7-nに含まれる情報が存在している場合には、事象待ち管理手段13により、その内の一つの事象待ちタスクが実行待ちタスク群6に移動される。逆に、実行中タスク5から、メッセージを要求する事象情報603が事象待ち管理手段13に伝達され、事象待ち管理手段13において、事象情報の内容、この場合にはメール・ボックスの番号が伝達されているものと判断されると、メール・ボックス12-1にメッセージがない場合には、実行中タスク5は事象待ちタスク群7に移動される。その際に、既にメッセージを待つタスクが、事象待ちタスク群7内に存在している場合には、当該事象待ちタスクに対応するタスク制御ブロック内にあるタスク接続アドレスが書き換えられる。

【0015】次に、イベント・フラグ12-2を利用する場合の事象待ち管理手段13の動作について説明する。まず、実行中タスク5から事象情報603が事象待ち管理手段13に伝達され、これを受けて事象待ち管理手段13において、当該事象情報603の内容、この場合にはイベント番号が伝達されているものと判断されると、事象待ち管理手段13により、イベント・フラグ12-2に伝達されたイベント・フラグの状態が、当該イベント・フラグ12-2に書き込まれる。更にまた、当該イベント・フラグ12-2内にある事象待ち行列、即ち事象待ちタスク群7のイベント待ちをしているタスクの情報の中に、事象待ちタスク7-1、7-2、……、タスク7-nに含まれる情報が存在している場合には、事象待ち管理手段13により、その内の一つ以上の事象待ちタスクが実行待ちタスク群6に移動される。逆に、実行中タスク5から、イベント・フラグがセットされるまで待つ事象情報603が事象待ち管理手段13に伝達され、事象待ち管理手段13において、事象情報の内容、この場合にはイベント番号が伝達されているもの

(5)

9

と判断されると、イベント・フラグ12-2に対象番号のイベント・フラグがセットされていない場合には、実行中タスク5は事象待ちタスク群7に移動される。その際に、既にイベントを待つタスクが、事象待ちタスク群7内に存在している場合には、当該事象待ちタスクに対応するタスク制御ブロック内にあるタスク接続アドレスが書き換えられる。

【0016】次ぎに、時間待ちタイマ12-3を利用する場合の事象待ち管理手段13の動作について説明する。まず、外部からのタイマ割り込みにより発生する事象制御情報602が事象待ち管理手段13に伝達され、これを受けて事象待ち管理手段13の制御により、時間待ちタイマ12-3内に時間待ちしているタスクが存在している場合には、時間待ちタイマ12-3に対応するタイム・カウンタがカウント・アップされる。そして、前記タイマ割り込みにより要求された時間の経過時点において、事象待ち管理手段13により、事象待ちタスク群7内において時間待ちをしているタスクの内の一つ以上の事象待ちタスクが実行待ちタスク群6に移動される。逆に、実行中タスク5から、時間待ち要求である事象情報603が事象待ち管理手段13に伝達され、事象待ち管理手段13において、事象情報の内容、この場合には時間情報が伝達されているものと判断されると、実行中タスク5は事象待ちタスク群7に移動される。その際に、既に時間待ちをするタスクが、事象待ちタスク群7内に存在している場合には、当該事象待ちタスクに対応するタスク制御ブロック内にあるタスク接続アドレスが書き換えられる。

【0017】次に、起床要求カウンタ12-4を利用する場合の事象待ち管理手段13の動作について説明する。まず、実行中タスク5から事象情報603が事象待ち管理手段13に伝達され、これを受けて事象待ち管理手段13において、当該事象情報603の内容、この場合には起床要求先タスク番号が伝達されているものと判断されると、事象待ち管理手段13により、起床要求カウンタ12-4に伝達された起床要求先タスク番号ごとに準備されている起床要求カウンタがインクリメントされる。更にまた、当該起床要求カウンタ12-4内に起床待ちしているタスクの情報の中、または時間待ちタイマ12-3内に時間待ちしているタスクの情報の中に、事象待ちタスク7-1、7-2、……、7-nに吹まれる情報が存在している場合には、事象待ち管理手段13により、その内の事象待ちタスクが実行待ちタスク群6に移動される。逆に、実行中タスク5から、起床待ち要求である事象情報603が事象待ち管理手段13に伝達され、事象待ち管理手段13において、事象情報の内容、この場合には起床要求がくるまで待つ情報が伝達されているものと判断されると、事象待ち管理手段13により、実行中タスク5は事象待ちタスク群7に移動される。その際に、既に起床待ちをするタスクが、事象待ち

10

タスク群7内に存在している場合には、当該事象待ちタスクに対応するタスク制御ブロック内にあるタスク接続アドレスが書き換えられる。

【0018】このように、従来のマルチプログラミング・システムにおいては、事象状態記憶領域12に格納されている事象情報が、種別（メッセージ、イベント、時間および起床要求等）ごとに分類されているために、事象待ち管理手段13による各記憶領域に対する管理手順が複雑化し、特に、当該マルチプログラミング・システムを、組み込みシステムに適用する場合においては、実行中タスク5、実行待ちタスク群6および事象待ちタスク群7等を含むプログラム記憶部50の記憶領域による制約により、事象待ち管理手段13により、全ての種別に対応する管理処理を行うことが困難になるという状態となっている。また、事象状態記憶領域12には、待ち行列を形成するためのタスク接続アドレスとして定義されるポインタが格納されているために、前記組み込みシステムに対応する場合には、タスク制御ブロック4および事象状態記憶領域12を含むデータ記憶部51に求められる記憶領域が大幅に増大するという問題が存在している。

【0019】図18(a)、(b)および(c)は、テープ録音式の留守番電話システムにおいて、OSを用いたマルチプログラミング・システムを組み込みシステムとして実現した場合の手順を示すタイミング図である。電話がかかってきた時に対応する留守番電話プログラムは、相手に対して応答メッセージを発声し、当該発声後に相手の声を録音する応答タスク30と、2msごとにテープ制御回路の動作状態の監視および制御を行うテープ制御ハンドラ31と、これらの応答タスク30およびテープ制御ハンドラ31の余り時間でキーおよび受話器の状態を監視するキー監視タスク32とが並行して動作する。これらのタスクの内、優先順位は、テープ制御ハンドラ31が一番高く、応答タスク30およびキー監視タスク32は優先順位が低く設定される。そして、それぞれのタスク内には、予め定められた順序に従って実行される処理（以下、シーケンスと云う）が用意されている。上記の各タスクは、事象待ち状態になるか、または優先順位が高い他のタスクの事象が成立した時には、他のタスクに実行権を譲渡し、疑似的にマルチ・タスク動作の状態となっている。

【0020】図18(a)、(b)および(c)は、それぞれ応答タスク30、テープ制御ハンドラ31およびキー監視タスク32のシーケンスの一例を示すタイミング図である。応答タスク30において実行権を譲渡するタイミングとして、電話がかかってくるのを待つタイミング $T_1$ と、応答メッセージの発声終了を待つタイミング $T_2$ と、相手の声を録音できる許容時間（30秒）の経過を待つタイミング $T_3$ と、録音した日付・時刻をテープに記録するのを待つタイミング $T_4$ とがある。シ

(6)

11

一ケンス30<sub>1</sub>においては、着信が検出された時点において事象待ち状態から復帰し、応答メッセージを発声して、発声終了待ちにおいて待ち状態に設定される。シーケンス30<sub>2</sub>においては、発声終了が検出された時点において事象待ち状態から復帰し、相手の声を録音した後30秒の時間待ちにて待ち状態にする。シーケンス30<sub>3</sub>においては、30秒が経過した時点において待ち状態から復帰し、日付・時刻が記録されて記録終了待ち状態に設定される。シーケンス30<sub>4</sub>においては、日付・時刻の記録が終了した時点において待ち状態から復帰し、着信待ちで待ち状態に設定される。シーケンス311からシーケンス317においては、2msの周期でテープの動作制御（録音、再生および停止）が行われる。またシーケンス321からシーケンス327においては、シーケンス30<sub>1</sub>からシーケンス30<sub>4</sub>の間における動作が行われていない時に動作が行われて、キーおよび受話器の状態が常時監視される。例えば、応答動作中に受話器が上げられた場合には、応答タスク30を強制終了させるシーケンスが含まれている。人が違和感を感じないように、少なくとも約100msごとに動作することが必要である。

【0021】図18におけるタイミングT<sub>1</sub>からタイミングT<sub>2</sub>に至る期間においては、応答メッセージの発声が終了したという事象の他に、相手が電話を切ったという事象が検出されないと、応答メッセージの発声が終了するまでの次のシーケンスに移行することができない。また、タイミングT<sub>2</sub>からタイミングT<sub>3</sub>の期間においては、相手の声を録音できる許容時間（30秒）が経過したという事象の他に、相手が電話を切ったという事象が検出されないと、相手が電話を切っても録音が継続して行われる状態となる。更に、タイミングT<sub>3</sub>からタイミングT<sub>4</sub>の期間においては、録音中にテープが一杯になったという事象が発生するという可能性がある。

【0022】このように、複数の事象を検出する必要がある場合には、従来のマルチプログラミング・システムにおける一つの対処方法としては、事象待ち状態に移行することなく可能な限り実行状態とし、優先順位の高いタスクのみを割り込みにより実行するという方法が採られている。図19(a)、(b)および(c)は、それぞれ応答タスク30、テープ制御ハンドラ31およびキー監視タスク32のシーケンスの一例を示すタイミング図であるが、この図19(a)、(b)および(c)には、上記の図18における電話を切ったという事象と、録音中にテープが一杯になったという事象とを加えることにより、複数の事象待ちを検出する必要がある場合の実現例におけるタイミング図が示されている。タイミングT<sub>1</sub>とタイミングT<sub>2</sub>とを検出するために、応答タスク30は、優先順位の高いテープ制御ハンドラ31が実行されていない時に動作する。本例においては、タイミングT<sub>1</sub>またはT<sub>2</sub>を事象待ちせずに検出しているため

12

に、当該応答タスク30よりも優先順位が低いキー監視タスク32が動作することができず、応答メッセージを再生して相手の声の録音が終了するか、または相手が電話を切るまで、受話器を合げたことを検出することができない。

【0023】複数の事象を検出することを必要とする場合の他の対処方法としては、それぞれの事象ごとに新たに事象待ちタスクを起動させて、そのタスクから応答タスク30に対して、起床要求またはイベント、メッセージ等を送信する方法がある。図20(b)および(c)は、テープ制御ハンドラ31およびキー監視タスク32のシーケンスの一例を示すタイミング図であるが、図20(a)には、上記の図18における電話を切ったという事象と、録音中にテープが一杯になったという事象とを加えることにより、複数の事象待ちを検出する必要のある場合の実現例におけるタイミング図が示されている。

【0024】この場合は、シーケンス30<sub>1</sub>においては、終了待ちシーケンス38、または切断待ちシーケンス43をタスクとして起動し、タイミングT<sub>2</sub>において、応答タスク30は、終了待ちシーケンス38、または切断待ちシーケンス43からの起床要求を受け、タイミングT<sub>3</sub>においては、応答タスク30は、経過待ちシーケンス39、切断待ちシーケンス43、または異常待ちシーケンス44から起床要求を受ける点において、図18のタイミング図とは異なっている。終了待ちシーケンス38は、応答メッセージの発声終了が検出された時点において、応答タスク30に対して起床要求をし、経過待ちシーケンス39は、相手の声を録音できる許容時間（30秒）が経過した時に、応答タスク30に対して起床要求をする。また、切断待ちシーケンス43は、相手が電話を切った時に応答タスク30に対して起床要求する。異常待ちシーケンス44は、録音中にテープが一杯になった時に、応答タスク30に対して起床要求をする。なお、応答待ちシーケンス38および経過待ちシーケンス39は、シーケンス30<sub>2</sub>において強制終了されるまで、事象待ちタスク群7、実行待ちタスク群6、または実行中タスク5に格納されており、経過待ちシーケンス39、切断待ちシーケンス43および異常待ちシーケンス44は、シーケンス30<sub>2</sub>において、強制終了されるまで、事象待ちタスク群7、実行待ちタスク群6、または実行中タスク5に格納されている。

【0025】本例においては、タイミングT<sub>2</sub>において、応答待ちシーケンス38が、応答タスク30に対して起床要求し、またタイミングT<sub>3</sub>において、経過待ちシーケンス39が、応答タスク30に対して起床要求をしている場合の動作例を示している。なお、本例においては、タスクの起動、強制終了および終了等の処理を頻繁に行う必要があるため、タスク管理が複雑化し、プログラム容量が増大する傾向となる。更に、複数のタスク



13

を起動させることに起因して、事象待ち行列に対応する格納領域が大量に消費されるという問題がある。

【0026】

【発明が解決しようとする課題】上述した従来のマルチプログラミング・システムにおいては、事象ごとに待ち行列が形成されているために、当該待ち行列のポインタ接続に伴い前記事象状態記憶領域を含む記憶領域が増大し複雑化するとともに、前記待ち行列を頻繁に生成／消滅させる際には、その使用頻度に応じて多大の処理実行時間が必要とするという欠点がある。

【0027】また、更に、待ち解除を行うシステムコール（特にイベント待ちを解除するシステムコール）の呼出しに際しては、多くのプログラム・ステップを要するという状態となり、プログラムをROMのようなファームウェアに具備し、且つ、記憶領域が限られたRAMに見られるような組み込みシステムに適用する場合においては事象の個数が制約され、大半近くがOS関連の処理およびデータに消費される結果となり、当該マルチプログラミング・システムの組み込みシステムに対する適用性に重大な支障を生じるという欠点がある。

【0028】更に、留守番電話システムのようにタイム・シーケンスが多い場合には、当該シーケンスの分岐において、複数の事象を同時に判断することが求められるが、そのためには実現する方法が複雑化し、逆にマルチプログラミング・システムを適用するという目的の一つである生産性を、却って悪化させる要因になるという欠点がある。

【0029】

【0030】

【課題を解決するための手段】本願発明のよる事象管理方式は、プログラムを細分化したタスクごとに設定される当該タスクの実行可能な条件（以下、事象と云う）の種別と、当該種別に対応する待ち内容と、条件が成立した結果からなる成立条件を格納する事象成立条件記憶手段と、使用頻度の高い特定事象の種別と当該種別に対応する待ちの内容とを格納する事象成立条件テーブルと、前記事象成立条件記憶手段に格納されている事象が成立した時点に対応する事象待ち解除条件を格納する事象待ち条件記憶手段と、前記タスクを制御するための制御情報を格納するタスク制御ブロックと、実行中のタスクにおいて事象待ちが発生した場合、該事象が事象成立条件テーブルにあるときは事象成立条件テーブルから登録内容を読み出して事象成立条件記憶手段に書きこみ、事象成立条件テーブルにないときは、発生した事象の成立条件を事象成立条件記憶手段に書きこむものであり、また、タスクの事象が発生するたびに、当該事象を前記事象成立条件記憶手段に格納されている事象成立条件と比較照合して、対象とする事象の成立の可否を検出するタスク付属事象管理手段と、実行中のタスクにおいて事象待ちが発生した場合、当該タスクを前記タスク制御ブ

(7)

14

ックを使用して事象待ち状態に遷移させ、また、当該事象待ち状態のタスクに対する前記事象待ち解除条件が成立した時点において、当該タスクを実行待ち状態に遷移させるタスク付属事象待ち管理手段と、前記タスク付属事象待ち管理手段により、事象待ち状態に遷移されたタスクを一時的に格納する事象待ちタスク記憶手段と、前記タスク付属事象待ち管理手段により、実行待ち状態に遷移されたタスクを一時的に格納する実行待ちタスク記憶手段と、所定のCPU（中央処理装置）において実行されているタスクを格納している実行中タスク記憶手段と、前記実行中のタスクと、前記実行待ちタスク記憶手段に格納されている実行待ちタスクとの内より一つのタスクを選択して、前記タスク制御ブロックを使用して実行状態に遷移させるタスク管理手段と、を少なくとも備えることを特徴としている。

【0031】

【実施例】次に、本発明について図面を参照して説明する。

【0032】図1は本発明を説明するための参考例を示す内部構成図である。図1に示されるように、本参考例は、タスク管理手段1と、タスク付属事象管理手段2と、タスク付属事象待ち管理手段3と、タスク制御ブロック4-1、4-2、……、4-nを含むタスク制御ブロック群4と、実行中タスク5と、実行待ちタスク6-1、6-2、……、6-nを含む実行待ちタスク群6と、事象待ちタスク7-1、7-2、……、7-nを含む事象待ちタスク群7と、割り込みハンドラ8と、事象成立条件記憶領域9-1、9-2、……、9-nを含む事象成立条件記憶領域群9と、事象待ち条件記憶領域10-1、10-2、……、10-nを含む事象待ち条件記憶領域群10とを備えて構成される。なお、上記の実行中タスク5、実行待ちタスク群6、事象待ちタスク群7および割り込みハンドラ8に含まれる情報は、プログラム記憶部50に格納されており、また、タスク制御ブロック群4、事象成立条件記憶領域群9および事象待ち条件記憶領域群10に含まれる情報は、データ記憶部51に格納されている。また、図2(a)および(b)は、本実施例における事象管理処理手順を示すフローチャートである。

【0033】図1において、事象成立条件記憶領域群9は、対応する実行中タスク5、実行待ちタスク群6、事象待ちタスク群7において使用される事象の内、現在、待ち状態にある事象の種別と、当該種別に対応する待ち内容と、条件が成立した結果とを、それぞれ実行中タスク5、実行待ちタスク群6、事象待ちタスク群7において使用される個数だけ格納する領域である。また、事象待ち条件記憶領域10は、事象成立条件記憶領域9に複数の条件が格納された時に、事象待ち状態から復帰するときの条件（ANDおよびOR等）と、復帰したときに、何れの事象成立条件によって復帰したかの情報を格

(8)

15

納しておく領域であり、タスク付属事象管理手段2は、事象成立条件記憶領域9の事象が発生したときに、当該事象成立条件記憶領域9の内容を更新する機能を有している。タスク付属事象待ち管理手段3は、事象待ち条件記憶領域10に示されている事象待ち解除条件が成立したときに、事象待ちタスク群7を実行待ちタスク群6に移行させる役割を果す。

【0034】まず、図1および図2(a)を参照して、実行中タスク5が、タスク付属事象管理手段2とタスク付属事象待ち管理手段3により、事象待ち状態に移行する処理手順について説明する。実行中タスク5において事象待ちが発生した場合にはタスク付属事象管理手段2に制御が移り、ステップA1においては、タスク付属事象管理手段2により、登録内容が事象成立条件記憶領域9に書き込まれて格納される。そしてステップA2に移行して、更に継続して書き込む登録内容が存在する場合には、ステップA1に戻り、再度当該登録内容が事象成立条件記憶領域9に書き込まれて格納される。この処理手順は、前記登録内容が無くなるまで継続して行われる。然る後に、登録内容が無くなった時点においてはタスク付属事象待ち管理手段3に制御が移り、ステップA3において、タスク付属事象待ち管理手段3により、実行中タスク5は事象待ちタスク群7に移動する。そして実行中タスク5の移動後においては、タスク管理手段1に制御が移り、このタスク管理手段1によりスケジューリングが行われ、実行待ちタスク群6の中から優先順位の高いタスクが、新たに実行中タスク5として設定される。

【0035】次に、図1および図2(b)を参照して、事象待ちタスク群7に含まれるタスクが、実行待ちタスク群5に移動する処理手順について説明する。外部回路またはCPUからのタイマ割り込みにより発生した事象制御情報602、実行中タスク5において発生した事象制御情報603、および割り込みハンドル8において発生した事象制御情報601を含む何れかの事象制御情報を受けて、タスク付属事象管理手段2に制御が移り、まずステップA4においては、タスク付属事象管理手段2により、発生した事象が事象成立条件記憶領域群9内に存在するか否かが判定される。当該事象が事象成立条件記憶領域群9内に存在しない場合には、タスク管理手段1に制御が戻り当該処理手順は終了する。また、当該事象が事象成立条件記憶領域群9内に存在する場合には、ステップA5において、当該事象の登録されている条件が成立しているか否かが判定され、成立していない場合にはタスク管理手段1に制御が戻り当該処理手順は終了する。また、登録されている条件が成立している場合には、ステップA6において、事象待ち条件記憶領域にある条件が成立しているか否かが判定され、成立していない場合にはタスク管理手段1に制御が戻り当該処理手順は終了となる。また、登録されている条件が成立してい

16

る場合には、ステップA7において、事象待ちタスク群7の中から、対象とするタスクが実行待ちタスク群6に移動する。そして、前記対象タスクの移動後においては、タスク管理手段1に制御が移り、このタスク管理手段1によりスケジューリングが行われ、実行待ちタスク群6の中から優先順位の高いタスクが、新たに実行中タスク5として設定される。

【0036】図3は、本参考例における事象成立条件記憶領域群9に含まれる事象成立条件記憶領域9-i(i=1, 2, …, n)、および事象待ち条件記憶領域群10に含まれる事象待ち条件記憶領域10-i(i=1, 2, …, n)の一構成例を示すブロック図である。事象成立条件記憶領域9-iには、それぞれ登録内容1、登録内容2、…、登録内容mが格納されている。これらの各登録内容には、それぞれ登録事象番号1、2、…、mを含む登録事象番号100が割り当てられており、システム・コールによって参照・登録するときに、この登録事象番号100によって登録内容が判別される。また、事象待ち条件記憶領域群10-iには、条件識別番号101と待ち解除結果102が格納されている。

【0037】事象待ち条件記憶領域群10は、事象成立条件記憶領域群9に登録されている事象が全て成立したとき(AND条件: 1)、何れかの事象が成立したとき(OR条件: 2)、または或る事象が成立したとき(単一条件: 0)に、待ち状態を復帰させるための条件を格納する条件識別番号102と、待ち状態が解除されたときに、最後に待ち状態を解除した登録事象番号を格納する待ち解除結果103とにより構成されている。

【0038】図4(a)、(b)、(c)、(d)、(e)および(f)は、本参考例における事象成立条件記憶領域群9の登録内容の一構成例を示すブロック図である。事象の種類としては、イベント待ち、メッセージ待ち、受信バッファ付きメッセージ待ち、範囲指定付きメッセージ待ち、時間待ち、および起床待ちを持った場合における例である。これらの事象は、それぞれ3バイトのデータにより構成されており、1バイト目100aの事象種別の識別番号に対応して、2バイト目100bと、3バイト目100cに待ち条件が格納されている。図4(a)に示されるように、1バイト目100aがイベント識別番号であった場合には、2バイト目100bにはフラグ・アドレスが格納され、3バイト目100cにはビット位置がその上位が格納され、現在のイベント状態がその下位に格納される。また図4(b)に示されるように、1バイト目100aがメッセージ識別番号であった場合には、2バイト目100bには待ちメッセージが格納され、3バイト目100cには受信メッセージが格納される。以下同様に、1バイト目100aが受信バッファ付きメッセージ識別番号であった場合には、2バイト目100bにはバッファ・アドレスが格納され、

(9)

17

3バイト目100cにはバッファ・サイズが格納される(図4(c)参照)。1バイト目100aが範囲指定付きメッセージ識別番号であった場合には、2バイト目100bには待ちメッセージの上限が格納され、3バイト目100cには待ちメッセージの下限が格納されるとともに、受信されたメッセージが重ね書きされて格納される(図4(d)参照)。1バイト目100aが時間識別番号であった場合には、2バイト目100bおよび3バイト目100cには残り待ち時間が格納される(図4

(e)参照)。また、この待ち時間は、OSのタイマ割り込みごとにカウント・ダウンされ、0になった時点において、当該待ち時間が経過したものと見做される。1バイト目100aが起床識別番号であった場合には、2バイト目100bおよび3バイト目100cには起床要求カウンタが格納される(図4(f)参照)。なお、前記起床要求カウンタが最大値の状態にある場合には、カウント・アップすることなく、起床要求のシステム・コールを発行したタスクにエラーを返す。本参考例においてはイベントに対する専用の領域を設けず、1ビットのフラグ領域を静的に配置し、そのフラグ領域のアドレスとビット位置とを登録内容として格納するものとする。

【0039】また、メモリ・マップドI/Oであった場合には、フラグを入力ポートおよび出力ポート等に割り当てることも可能である。更に、イベントの前の状態とアクティブを指定することにより、エッジ検出によりタスクを起動することも可能となる。これにより、従来行われているように、ポートの状態を加工して、イベント・フラグをセット/リセットする必要がなくなる。ただし、イベント・フラグを入力ポート等のOS管理外において変化する領域、即ち、システム・コールを使用することなくフラグが変化する領域に割り当てた場合には、OSのタイマ割り込みにより検出される。メッセージについては、メール・ボックスを設けず、当該メッセージは直接タスクに対して送信される。メッセージ長を1バイトにすることにより、ポインタ等を格納するときのオーバ・ヘッドが軽減され、指定メッセージまたは範囲指定によりメッセージの受信を待つことにより、タスクが実行状態となる頻度が減殺される。また、従来は、時間待ち中に起床要求のシステム・コールが発行されたときには時間待ちが解除されているが、本参考例においては、起床待ちを設定しない限り時間待ちが解除されないようにすることもできる。

【0040】図5は、本参考例において、外部回路またはCPUからのOSタイマ割り込みを受けて、タスク付属事象待ち管理手段3が起動されるまでの処理手順を示すフローチャートの一例であり、図6および図7は、タスク付属事象管理手段2のシステム・コールを受けて、タスク付属事象待ち管理手段3が起動されるまでの処理手順を示すフローチャートの一例である。

【0041】図5において、OSタイマ割り込み200

18

は、一定時間ごとに、例えば1msごとにタスク等の処理中に割り込まれる。これを受けて、ステップ204においては、時間待ちのタスクが存在するか否かが判定され、存在しない場合にはステップ208に移行し、存在する場合にはステップ205においてタイムカウントされて、時間待ちを行っているタスクの事象内容の残り時間がデクリメントされる。次いで、ステップ206においては、タイムがUPであるか否かが判定されて、UPでない場合にはステップ208に移行し、UPである場合には、ステップ207において時間待ちが解除されて、再度ステップ204に戻り以降の処理手順が繰返して実行される。

【0042】上記のステップ204およびステップ206の処理手順において判定条件が満たされない場合にはステップ208に移行するが、ステップ208、209および210の処理手順は、入力ポート等のOS管理外において変化する領域にイベント・フラグを割り当てている場合において、イベントの成立を管理する際の処理手順である。この場合、OSのタイマ割り込み周期よりも短い時間で変化するときには、上記のステップ208、209および210の処理手順を更に短い周期で動作させることにより、当該処理の精度を向上させることができる。まずステップ208においては、イベント待ちのタスクが存在するか否かが判定されて、イベント待ちのタスクが存在しない場合には、これを受けてタスク付属事象待ち管理手段3が起動される。またイベント待ちのタスクが存在する場合には、ステップ209において当該イベントが満たされているか否かが判定され、満たされていない場合には、これを受けてタスク付属事象待ち管理手段3が起動される。またイベントが満たされている場合には、ステップ210において、当該イベント待ちが解除されてステップ208に戻り、以降の処理手順が継続して実行される。

【0043】図6においては、メッセージを送信するシステム・コール201を受けて、まずステップ211において送信先のタスクが存在するか否かが判定され、当外送信先タスクが存在しない場合には、ステップ217に移行して、前記システム・コールの発行先に対して送信エラーが返送されて復帰する。また、ステップ211において送信先のタスクが存在する場合には、次のステップ212に移行する。なお、以降のステップ212、213および214の判断処理においても判定条件が満たされない場合には、何れの場合においても同様にステップ217に移行して、前記システム・コールの発行先に対して送信エラーが返送されて復帰する。ステップ212においては送信タスクがメッセージ待ちしているか否かが判定されて、メッセージ待ちしている場合には、次のステップ213において、送信メッセージと待ちメッセージが同一であるか否かが判定されて、同一である場合には、ステップ214において、バッファに空き領

(10)

19

域が存在するか否かが判定され、バッファに空き領域が存在する場合には、ステップ215に移行して、当該メッセージは送信先の受信バッファに書き込まれる。次いで、ステップ216においてはメッセージ待ちが解除され、これを受けて、タスク付属事象待ち管理手段3が起動される。

【0044】次に、図7(a)においては、イベントを設定するシステム・コール202を受けて、ステップ218においては、同じイベントのフラグ・アドレスを持つタスクが存在するか否かが判定され、当該タスクが存在しない場合には、ステップ222に移行して、イベント・エラーがシステム・コールの発行先に返送されて復帰状態となる。また、ステップ218において同じイベントのフラグ・アドレスを持つタスクが存在する場合には、ステップ219において当該イベントが満たされているか否かが判定されて、満たされていない場合にはステップ221に移行し、満たされている場合にはステップ220において、イベント待ちが解除される。ステップ221においては、同じイベントを持つタスクが存在するか否かが判定されて、存在する場合にはステップ219に戻り、以降の処理手順が継続して実行される。また当該タスクが存在しない場合には、これを受けて、タスク付属事象待ち管理手段3が起動される。

【0045】また、図7(b)においては、起床を要求するシステム・コール203を受けて、ステップ223において、起床要求先のタスクが存在するか否かが判定され、存在しない場合にはステップ226に移行して、起床要求エラーがシステム・コールの発行先に返送されて復帰する。またステップ223において前記起床要求先のタスクが存在する場合には、ステップ224において、起床要求カウンタが最大値であるか否かが判定されて、最大値である場合にはステップ226に移行して、起床要求エラーがシステム・コールの発行先に返送されて復帰する。また最大値ではない場合には、ステップ225において起床要求カウンタがインクリメントされ、次いでステップ227において起床要求待ちが解除される。そして、これを受けてタスク付属事象待ち管理手段3が起動される。

【0046】図8は、本参考例におけるタスク付属事象待ち管理手段3における処理手順を示すフローチャートの一例である。まず、ステップ228において、事象待ちタスク群7内の事象待ち行列にタスクが存在するか否かが判定される。当該事象待ち行列にタスクが存在しない場合には、タスク管理手段1に制御が移り、タスク付属事象待ち管理手段3の動作は終了する。また当該事象待ち行列にタスクが存在する場合には、ステップ229において、事象待ちが、待ち条件のとうりに解除されているか否かが判定される。ステップ229において解除されていない場合には、タスク管理手段1に制御が移り、タスク付属事象待ち管理手段3の動作は終了する。

20

また、待ち条件のとうりに解除されている場合には、ステップ230において、事象待ちスタック群7内の事象が解除されている事象待ち行列が、実行待ちスタック群6内の実行待ち行列に移動する。このことは、既に、前述の図5、図6および図7に示されるフローチャートにより明らかなところである。但し、この場合、ステップ229の判断処理において、条件識別番号により示される事象待ち条件が満たされている場合においてのみ、ステップ230においては、前記事象待ち行列が前記実行待ち行列に移動するものとする。次いで、ステップ231においては、解除された事象の登録事象番号が待ち解除結果に書き込まれる。また、実行待ちタスク群6に移動したタスクは、タスク管理手段1において必要に応じて実行状態に遷移する。

【0047】図9は、本参考例における事象待ちに関するシステム・コールの一例を示す図である。システム・コール `set-phe` および `wai-phe` は、図3に示される登録内容1、登録内容2、……、登録内容nを含む登録内容100に指定された事象の種別と事象の内容を設定するシステム・コールであり、この `set-phe` により複数の条件が指定され、また `wai-phe` により最後の条件および条件識別番号102が指定されて、事象待ちの状態となる。また、`clr-phe` は、上記の `set-phe` または `wai-phe` により設定された事象の内の不要となった登録内容100を解除するシステム・コールであり、`pol-phe` は、事象待ちが解除されたときに、どの事象により解除されたかを、各登録内容100に指定されている登録事象番号を返すことにより知らせるシステム・コールである。そして `get-phe` は、各事象登録内容100の現在の状態を取り出すシステム・コールである。例えば、メッセージ待ちであれば受信メッセージを返し、時間待ちであれば残り時間を返すというように動作する。なお、本実施例においては、システム・コール名の最後の3文字には、現象を意味する英文字の先頭の `phe` を用いている。

【0048】なお、本発明は、種々の組み込みシステムに対して適用されるが、一例として、本参考例が留守番電話システムに適用される場合には、プログラム記憶部50には、留守番電話システム全体を制御する情報が格納され、特に、実行中タスク5、実行待ちタスク群6および事象待ちタスク群7には、留守番電話システムの操作情報が格納される。割り込みハンドラ8には、常時監視の必要のある（例えば、10msごとに監視する）外部情報が格納される。また、データ記憶部51には、留守番電話システムの実行状態が格納される。特に、事象成立条件記憶領域群9に対しては、事象待ちタスク群7に遷移タイミング情報を少なくとも一つ以上格納し、事象待ち条件記憶領域群10には、二つ以上の遷移タイミングの組合わせ情報が格納される。そして、タスク制御

(11)

21

ブロック4には、実行中タスク5、実行待ちタスク群6および事象待ちタスク群7の実行状態を制御する情報が格納される。図10は、本発明の第1の実施例を示す内部構成図である。図10に示されるように、本実施例は、タスク管理手段1と、タスク付属事象管理手段2と、タスク付属事象待ち管理手段3と、タスク制御ブロック4-1、4-2、……、4-nを含むタスク制御ブロック群4と、実行中タスク5と、実行待ちタスク群6-1、6-2、……、6-nを含む実行待ちタスク群6と、事象待ちタスク群7-1、7-2、……、7-nを含む事象待ちタスク群7と、割り込みハンドラ8と、事象成立条件記憶領域9-1、9-2、……、9-nを含む事象成立条件記憶領域群9と、事象待ち条件記憶領域10-1、10-2、……、10-nを含む事象待ち条件記憶領域群10と、事象成立条件テーブル11とを備えて構成される。なお、上記の実行中タスク5、実行待ちタスク群6、事象待ちタスク群7、割り込みハンドラ8および事象成立条件テーブル11に含まれる情報は、プログラム記憶部50に格納されており、また、タスク制御ブロック群4、事象成立条件記憶領域群9および事象待ち条件記憶領域群10に含まれる情報は、データ記憶部51に格納されている。図1の図10との対比により明らかなように、本実施例と前述の参考例との相違点は、本実施例においては、プログラム記憶部50の内部に事象成立条件テーブル11が新たに付加されていることである。また、図11(a)および(b)は、本実施例における事象管理処理手順を示すフローチャートである。

【0049】図10において、新たに付加された事象成立条件テーブル11は、図1の前述の参考例における事象の種別と、当該種別に対応する待ち内容の内、頻繁に使用するものを定数テーブルとして格納したものであり、組み込みシステムにおいてはROM領域に配置される。これによる前述の参考例との動作上の差異は、実行状態のタスク5において事象待ちを要求した時に、事象成立条件テーブル11における事象成立条件テーブル番号が指定されていれば、タスク付属事象管理手段3により、登録内容が事象成立条件として登録される。以後、前述の参考例の場合と同様に動作が行われる。図3に示される事象成立条件記憶領域群9に格納されている登録内容100を含む事象を予めテーブル化して、システム・コールでの条件記述をテーブル番号により記述して簡略化することにより、プログラム・ステップ数を減少させることが可能となる。

【0050】図11(a)は、実行中タスク5が、タスク付属事象管理手段2とタスク付属事象待ち管理手段3により事象待ちになる処理手順を示しており、また図11(b)は、事象待ちタスク群7のタスクが実行待ちタスク群6に移動する処理手順を示している。本実施例における処理手順と、前述の参考例における処理手順との

22

相違点は、図11(a)と、図2(a)との対比により明らかなように、本実施例においては、図11(a)にけるステップA8とステップA9において、事象成立条件を事象成立条件テーブル11から読み出して、事象成立条件記憶領域9に書き込むことである。また事象成立条件テーブル11に、予め複数条件の事象を登録しておき、ステップA9以降においてはステップA2における処理を行うことなく、ステップA3に移行することも考えられる。それ以外のステップA1、ステップA2、ステップA3、ステップA4、ステップA5、ステップA6およびステップA7を含む各処理手順にける処理内容については、前述の参考例の場合と同様である。

【0051】図12は、本実施例における事象成立条件テーブル11の構成例を示すブロック図である。図12において、登録事象数300は、登録内容1、……、登録内容nを含む登録内容301の個数を示している。これらの登録内容301は、図3に示される事象成立条件記憶領域9に設定される事象の種別と事象の条件とを初期値としてテーブル化されている。図9に示されたシステム・コールset-pheまたはwai-pheにより、登録事象テーブル番号が指定された時には、対応する登録内容301が取り出され、事象成立条件記憶領域群9に書き込まれる。また、図13(a)、(b)、(c)、(d)、(e)および(f)は、本実施例における事象成立条件テーブル11の登録内容の構成例を示すブロック図であり、図4に示される登録内容と同等ではあるが、ROM領域であるために、図13(b)の3バイト目301cが空き領域となっており、図13(d)の3バイト目301cには、待ちメッセージ下限のみが格納され、更に、図13(b)の3バイト目301cには、起床要求カウンタ初期値が格納されており、メッセージ受信領域およびイベント状態領域等が考慮されていない点に差異がある。

【0052】なお、前述の参考例の場合と同様に、本実施例が留守番電話システムに適用される場合には、プログラム記憶部50には、留守番電話システム全体を制御する情報が格納され、特に、実行中タスク5、実行待ちタスク群6および事象待ちタスク群7には、留守番電話システムの操作情報が格納される。事象成立条件テーブル11には、事象成立条件の内、留守番電話システムにおいて頻繁に使用される事象の組み合わせ情報が格納される。割り込みハンドラ8には、常時監視の必要のある(例えば、10msごとに監視する)外部情報が格納される。また、データ記憶部51には、留守番電話システムの実行状態が格納される。特に、事象成立条件記憶領域群9に対しては、事象待ちタスク群7に遷移タイミング情報を少なくとも一つ以上格納し、事象待ち条件記憶領域群10には、二つ以上の遷移タイミングの組み合わせ情報が格納される。そして、タスク制御ブロック4には、実行中タスク5、実行待ちタスク群6および事象待

(12)

23

ちタスク群7の実行状態を制御する情報が格納される。

【0053】

【発明の効果】以上説明したように、本発明は、タスク付属事象管理手段とタスク付属事象待ち管理手段とを備えることにより、事象待ちタスク群に含まれる待ち行列を簡略化することが可能となり、事象待ちタスクを頻繁に生成・起動・終了・消去する際における実行時間と、所要のプログラム・ステップとを削減することができるという効果がある。

【0054】更に、留守番電話のように、シーケンスと条件判断処理の多いシステムに対して本発明を適用することにより、極めて簡単に複数事象を判断することが可能となるために不必要なタスクが減殺されるとともに、シーケンスと条件判断との関係を考慮した当該システムの構造化設計が容易となり、当該システムの生産性と保守性を向上させることができるという効果がある。

【図面の簡単な説明】

【図1】本発明を説明するための参考例を示すブロック図である。

【図2】参考例における処理手順のフローチャートを示す図である。

【図3】参考例の事象成立条件記憶領域群と事象待ち条件記憶領域群を示すブロック図である。

【図4】参考例の事象成立条件記憶領域群における登録内容の構成例を示す図である。

【図5】参考例のタスク付属事象管理手段における処理手順のフローチャートを示す図である。

【図6】参考例のタスク付属事象管理手段における処理手順のフローチャートを示す図である。

【図7】参考例のタスク付属事象管理手段における処理手順のフローチャートを示す図である。

【図8】参考例のタスク付属事象待ち管理手段における処理手順のフローチャートを示す図である。

【図9】参考例における事象待ちに関するシステム・コールの一例を示す図である。

【図10】本発明の第1の実施例を示すブロック図である。

【図11】第1の実施例における処理手順のフローチャートを示す図である。

【図12】第1の実施例の事象成立条件記憶領域群と事象待ち条件記憶領域群を示すブロック図である。

【図13】第1の実施例の事象成立条件記憶領域群における登録内容の構成例を示す図である。

【図14】従来例を示すブロック図である。

【図15】従来例における処理手順のフローチャートを示す図である。

【図16】従来例における待ち行列形成例を示す図である。

【図17】従来例における制御構造を示すブロック図である。

24

【図18】従来例を留守番電話に適用した場合のタイミング図である。

【図19】従来例において、複数事象待ちを実現した場合のタイミング図である。

【図20】従来例において、複数事象待ちを実現した場合のタイミング図である。

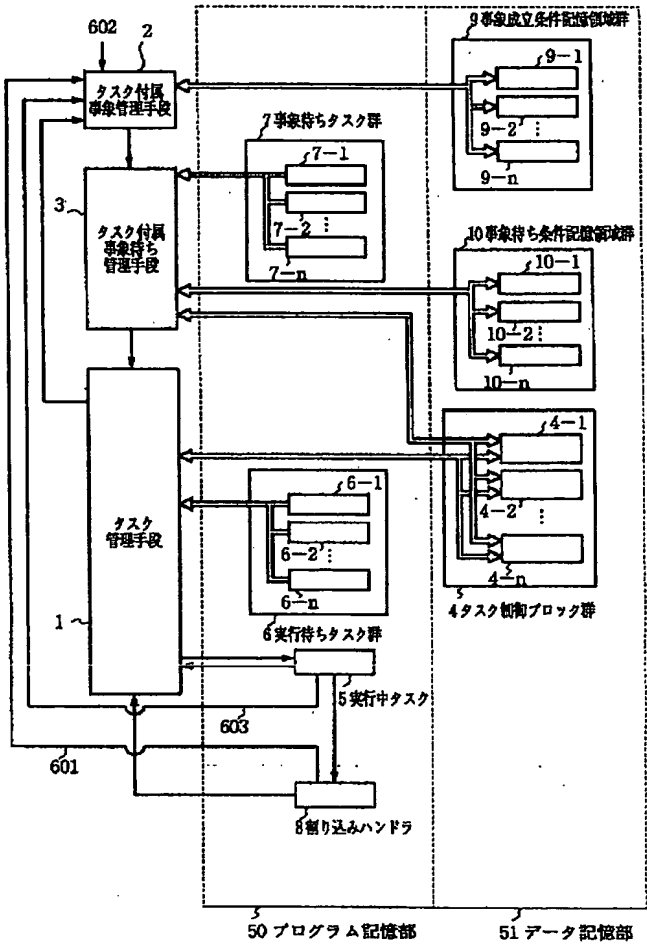
【符号の説明】

- 1 タスク管理手段
- 2 タスク付属事象管理手段
- 3 タスク付属事象待ち管理手段
- 4 タスク制御ブロック群
- 4-1～4-n タスク制御ブロック
- 5 実行中タスク
- 6 実行待ちタスク群
- 6-1～6-n 実行待ちタスク
- 7 事象待ちタスク群
- 7-1～7-n 事象待ちタスク
- 8、22、31 割り込みハンドラ
- 9 事象成立条件記憶領域群
- 9-1～9-n 事象成立条件記憶領域
- 10 事象待ち条件記憶領域群
- 10-1～10-n 事象待ち条件記憶領域
- 11 事象成立条件テーブル
- 12 事象状態記憶領域
- 12-1、12-2 メール・ボックス
- 12-3 時間待ちタイマ
- 12-4 起床要求カウンタ
- 13 事象待ち管理領域
- 14、201～203 システムコール
- 30 応答タスク
- 31 テープ制御ハンドラ
- 32 キー監視タスク
- 301～307、311～317、321～327 シーケンス
- 38 音声終了待ちシーケンス
- 39 経過待ちシーケンス
- 43 切断待ちシーケンス
- 44 異常待ちシーケンス
- 45 実行待ち行列
- 46 時間待ち行列
- 47 イベント待ち行列
- 48 メッセージ待ち行列
- 50 プログラム記憶部
- 51 データ記憶部
- 100 登録内容1～登録内容m
- 100a、300a 1バイト目
- 100b、300b 2バイト目
- 100c、300c 3バイト目
- 101 条件識別番号
- 102 待ち解除結果

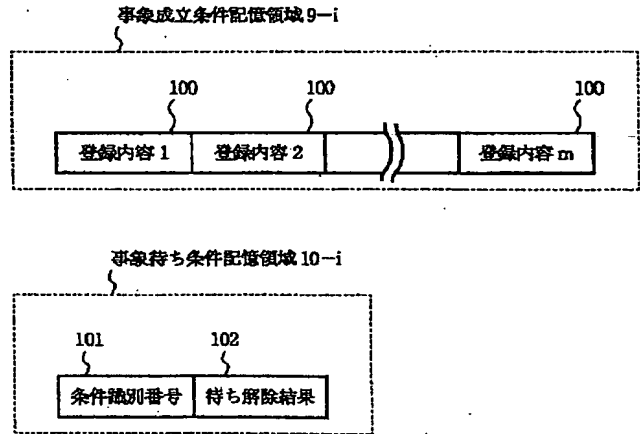
(13)

204~233、A1~A7、B1~B4 ステップ  
300 登録事象数  
301 登録内容1~登録内容m

【図1】

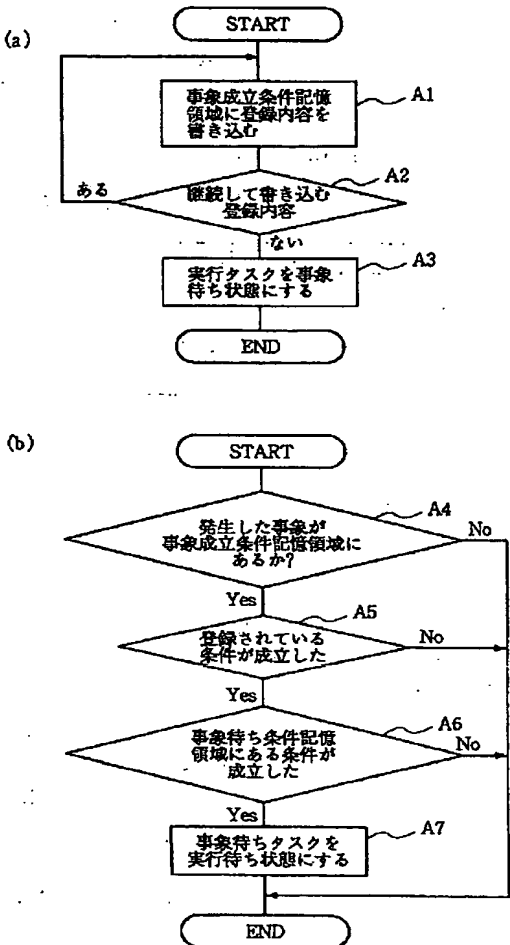


【図3】



26  
400、4011~4013、402、4031、4032、404~406 ポインタ  
601~603 事象制御情報

【図2】

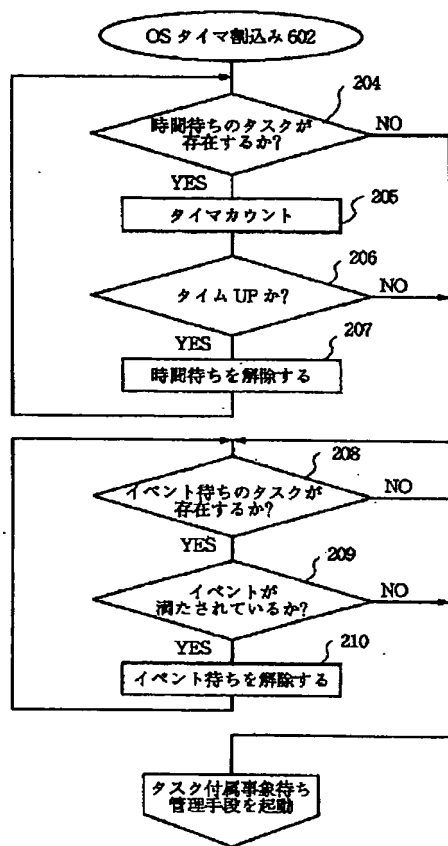


【図4】

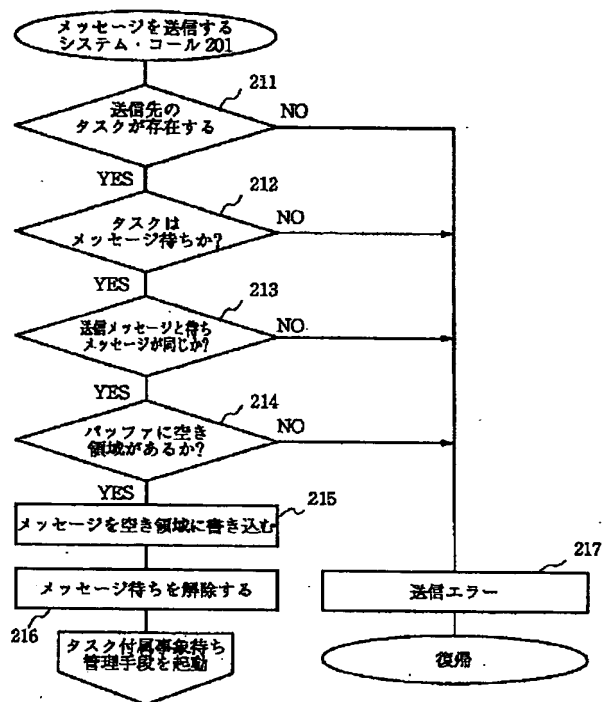
	100a 1バイト目	100b 2バイト目	100c 3バイト目
(a) イベント待ち	イベント識別番号	フラグ・アドレス	ビット位置 イベント状態
(b) メッセージ待ち	メッセージ識別番号	待ちメッセージ	受信メッセージ
(c) 受信バッファ付き メッセージ待ち	受信バッファ付き メッセージ識別番号	バッファ・アドレス	バッファ・サイズ
(d) 範囲指定付き メッセージ待ち	範囲指定付き メッセージ識別番号	待ちメッセージ上限	待ちメッセージ下限 / 受信メッセージ
(e) 時間待ち	時間識別番号	残り待ち時間	
(f) 起床待ち	起床識別番号	起床要求カウンタ	

(14)

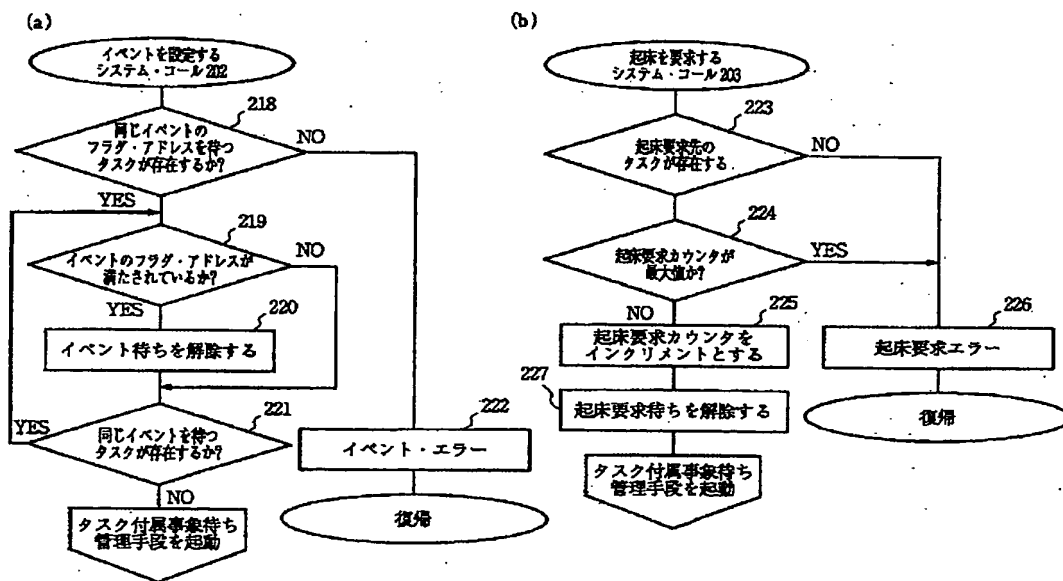
【図5】



【図6】



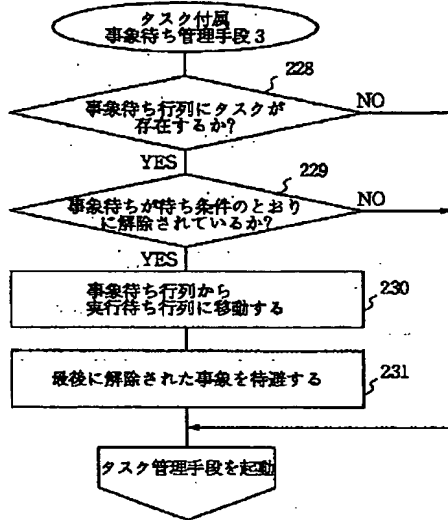
【図7】





(15)

【図8】

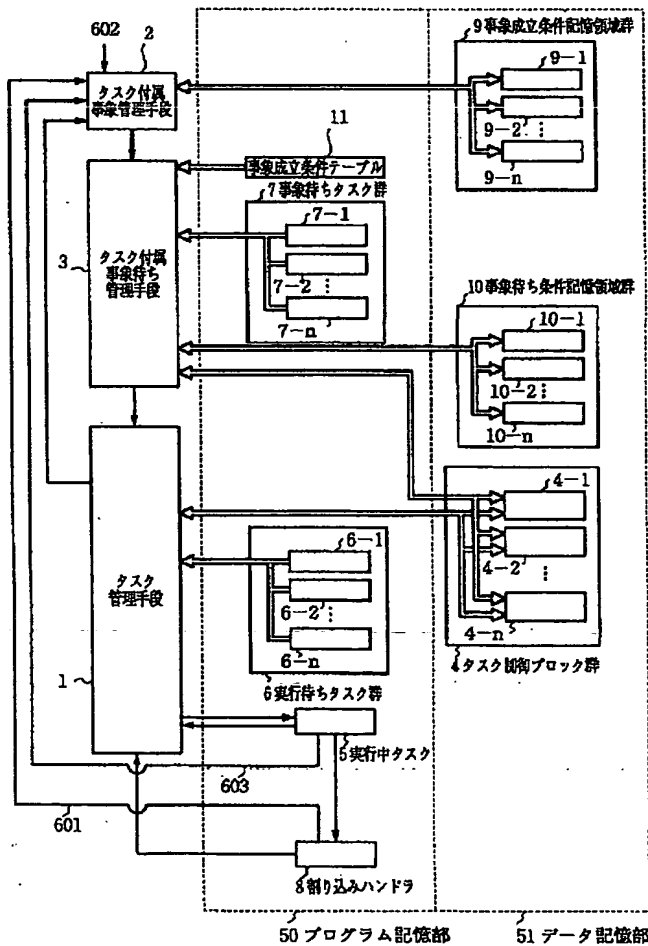


【図9】

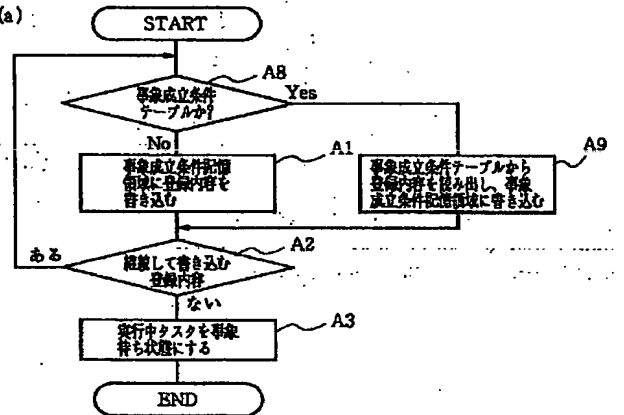
変数名	動作説明
set_flg	指定した事象番号に事象待ち条件を設定する。 (待ち条件：イベント待ち、メッセージ待ち、時間待ち、起床待ち)
wait_flg	指定した事象番号に事象待ち条件を設定し、待ち状態に移移する。 set_flg で設定した事象との組み合わせで待ち状態にする。
clr_flg	指定した事象番号の事象待ち条件を解除する。
get_flg	事象待ちが解除されたとき、どの事象番号の事象によって解除されたかを検出する。
set_flg	指定した事象番号の事象待ち状態を取り出す。ただし、以下の待ち条件によって戻す値の意味が異なる。 ①イベント待ち：イベントの状態（セット、リセット） ②メッセージ待ち：受信メッセージ、または、メッセージなし ③時間待ち：残り時間 ④起床待ち：残り起床要求数

【図11】

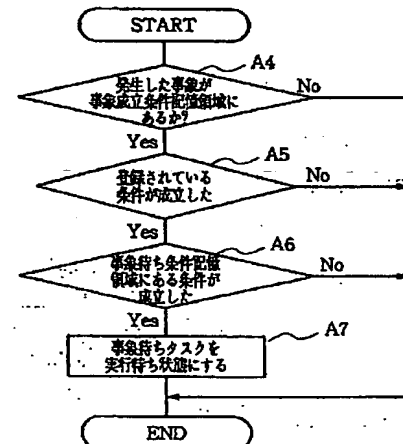
【図10】



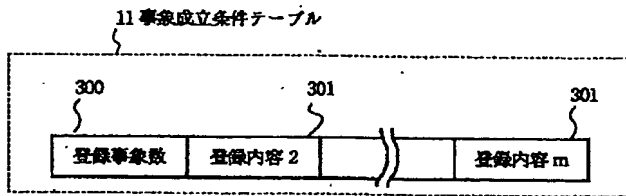
(a)



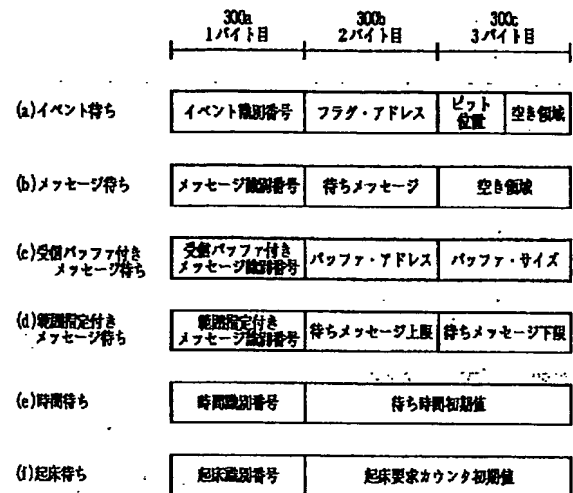
(b)



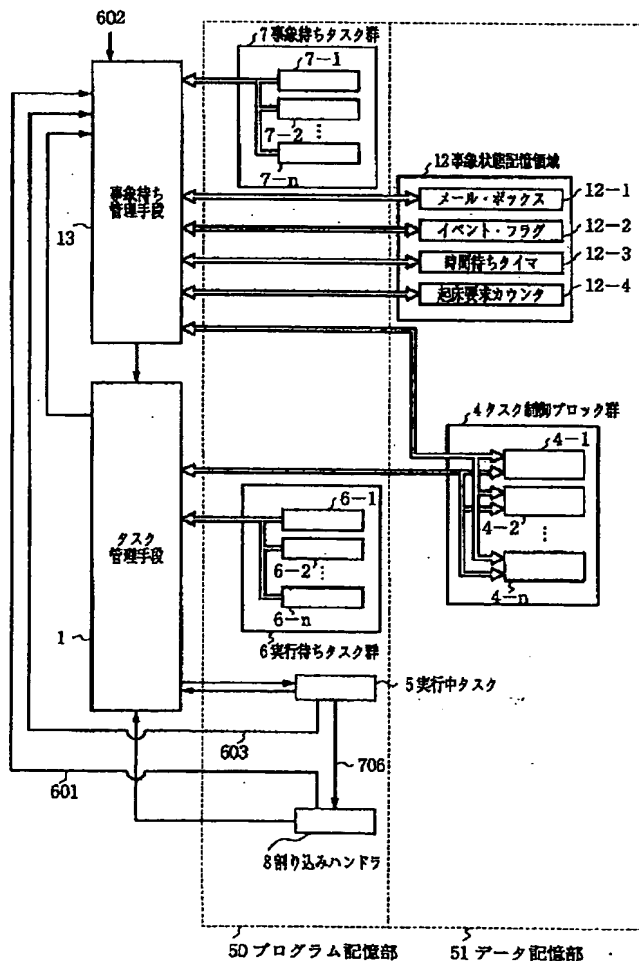
【図 12】



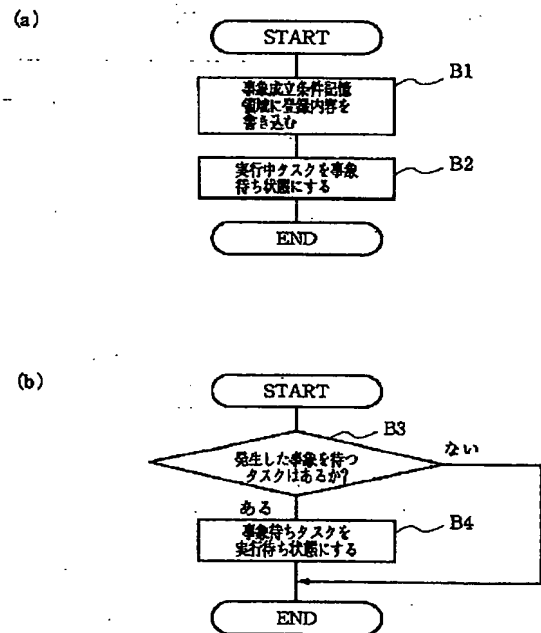
【图 1 3】



【図 14】

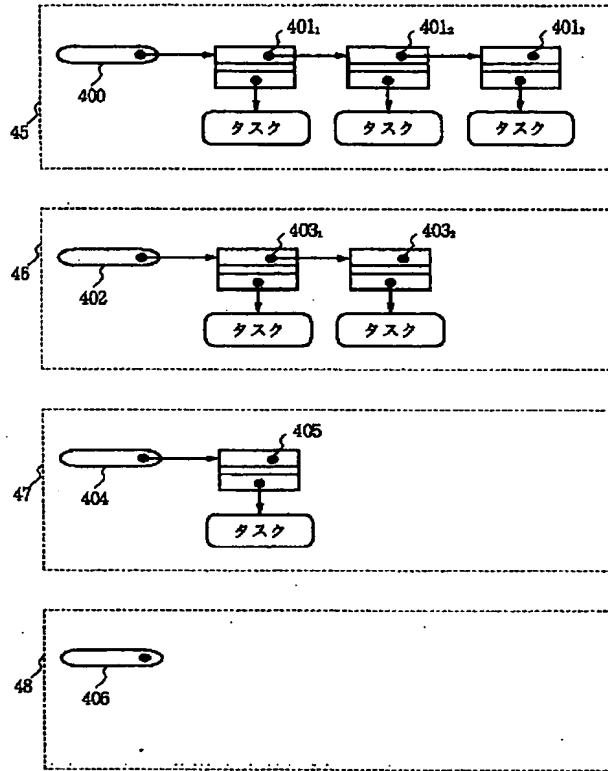


【図 15】



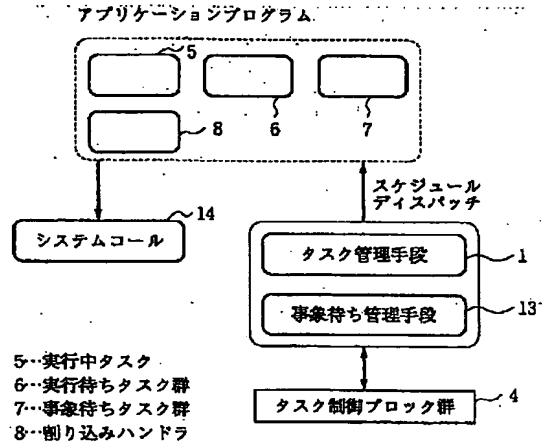
(17)

【図16】



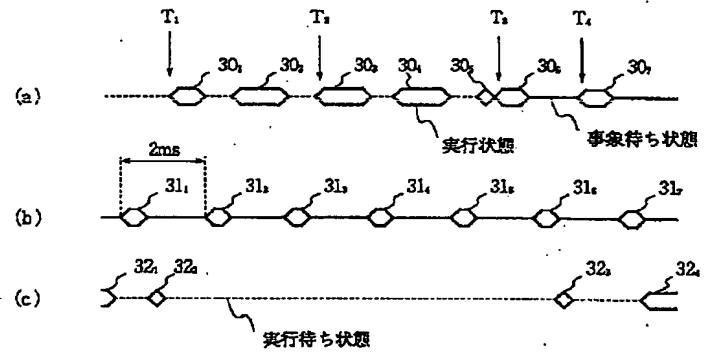
45...実行待ち行列  
46...時間待ち行列  
47...イベント待ち行列  
48...メッセージ待ち行列

【図17】

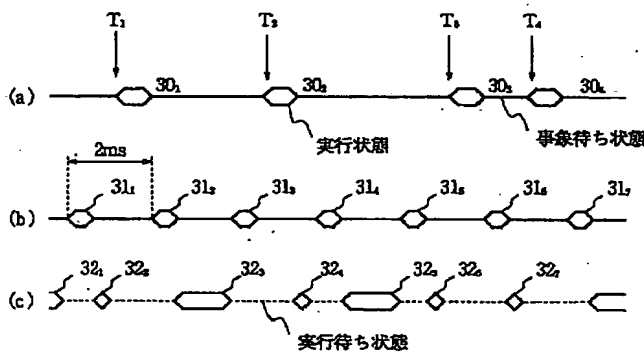


5...実行中タスク  
6...実行待ちタスク群  
7...事象待ちタスク群  
8...割り込みハンドラ

【図19】

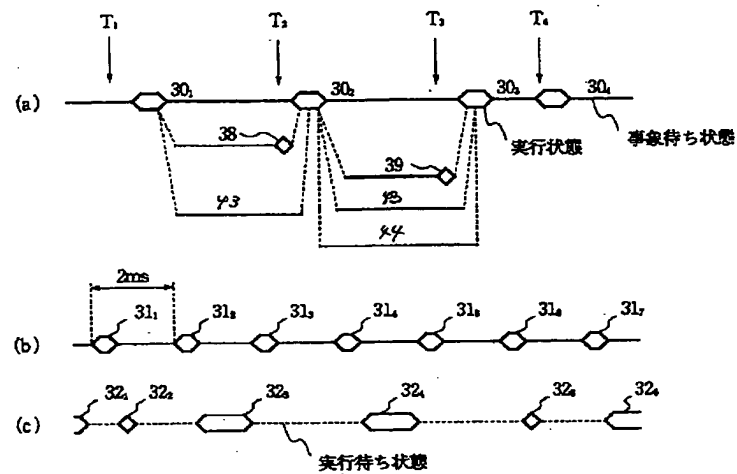


【図18】



(18)

【図20】



フロントページの続き

(56) 参考文献 特開 平4-287233 (JP, A)  
 特開 平1-248240 (JP, A)  
 特開 平3-25624 (JP, A)  
 特開 平6-35724 (JP, A)  
 特開 平5-80972 (JP, A)  
 特開 昭59-99531 (JP, A)  
 発明協会公開技報・公技番号93-  
 25932  
 発明協会公開技報・公技番号93-  
 26242  
 発明協会公開技報・公技番号93-2017

(58) 調査した分野 (Int. Cl. 7, DB名)  
 G06F 9/46 340

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**